Research Article

New Adaptive Algorithms for GOP Size Control with Return Channel Suppression in Wyner-Ziv Video Coding

Charles Yaacoub,1, 2 Joumana Farah,1 and Béatrice Pesquet-Popescu2

1 Engineering Department, Faculty of Sciences and Computer Engineering, Holy-Spirit University of Kaslik, P.O. Box 446, Jounieh, Keserwan, Mount Lebanon, Lebanon
2 Signal and Image Processing Department, TELECOM ParisTech, 46 Rue Barrault, 75634 Paris Cedex 13, France

Correspondence should be addressed to Charles Yaacoub, charlesyaacoub@usek.edu.lb

Received 28 June 2008; Revised 8 September 2008; Accepted 22 October 2008

1. Introduction

Distributed source coding [1–20] has recently become a topic of great interest for the research community, especially in the world of video communications. In traditional video coding techniques, such as MPEG or H.26x, motion estimation is performed at the encoder side, which yields very complex encoders, but simple decoders. This is suitable for applications where a video sequence is encoded once and decoded several times, such as video broadcasting or video streaming on demand. A simple decoder is desired in this case to allow low-cost receivers for the end users.

On the other hand, some applications require simple encoders. Distributed Video Coding (DVC) was introduced [7, 8] to permit low-complexity encoding for small power-limited and memory-limited devices, such as camera-equipped mobile phones or wireless video sensors, by moving the computation burden from the encoder side to the decoder. Increased decoding complexity can be tolerated in this case since, in such applications, the decoder is usually located in a base station with sufficient resources.

It is known from information theory that, given two statistically dependent sources X and Y, each source can be independently compressed to its entropy limit, H(X) and H(Y), respectively. However, by exploiting the correlation statistics between these sources, X and Y can be jointly compressed to the joint entropy H(X, Y). This results in a more efficient compression since H(X, Y) ≤ H(X) + H(Y). The idea behind DVC goes back to the 1970s when Slepian and Wolf [21] proved that, if the source Y is compressed to its entropy limit H(Y), X can be transmitted at a rate very close to the conditional entropy H(X|Y), provided that Y is available at the receiver as side information for decoding X. Since H(X, Y) = H(Y) + H(X|Y), X and Y can be independently encoded and jointly decoded without any loss in the compression efficiency, compared to the case where both sources are jointly encoded and decoded. The application of this concept to lossy source coding is known as the Wyner-Ziv coding [22].

In practical DVC systems, a subset of frames, known as key frames, is usually compressed using traditional intracoding techniques. One or more frames following each key frame, known as Wyner-Ziv (WZ) frames, are then...
compressed by appropriate puncturing of the parity bits at the output of a channel coder. At the receiver side, previously decoded (key or WZ) frames are interpolated to generate the necessary side information for the decoding process.

The first practical DVC systems appeared in 2002, when Puri and Ramchandran [7] proposed a block-based codec using syndromes, and Aaron et al. [8] proposed a frame-based codec using turbo codes. The frame-based approach has gained a greater interest in the research community. However, it still suffers from several weaknesses that limit its use in real-life applications.

One of the main drawbacks of current DVC systems is the use of a feedback channel (FC) [11] to allow flexible rate control and to ensure successful decoding of WZ frames. The FC is not suitable for real-time systems (e.g., broadcasting applications) due to transmission delay constraints. Additionally, in multiuser applications with rate constraints, the application of WZ coding becomes impractical because of the difficulty of implementing appropriate rate allocation algorithms. Furthermore, since several decoding runs are required to successfully recover a WZ frame, the FC imposes instantaneous decoding in the receiver. For all these reasons, the introduction of new techniques for estimating the necessary bit rate to successfully decode each WZ frame becomes crucial. In fact, the problem of the return channel in DVC has rarely been targeted in the literature. Artigas and Torres [12] and Morbée et al. [13] proposed techniques that rely on performance tables used by the encoder to predict the compression level of each particular frame. Kubasov et al. proposed in [18] an encoder rate control technique that reduces the use of the feedback channel. Transform domain WZ rate control algorithms were introduced in [19] (for a DCT-based WZ codec) and [20] (for a wavelet-based WZ codec). However, these studies do not take into account the rate constraints in limited-bandwidth applications. Besides, the influence of the channel impairments on the proposed rate allocation techniques is not considered. In [6, 14, 15], we proposed a novel technique for the removal of the feedback channel in DVC systems, using an analytical approach based on entropy calculations. Designed for a multiuser scenario, the proposed technique takes into account the amount of motion in the captured video scene as well as the transmission channel conditions for every user, in order to allocate unequal transmission rates among the different users.

Another drawback of current DVC systems is that the quality of the generated side information greatly affects the system's performance. Even though the interpolation algorithm used at the decoder strongly influences the side information quality, key frames are essential components in the interpolation process and thus, having high-quality key frames is crucial. Therefore, a very high peak signal-to-noise ratio (PSNR) is desired (for the key frames) in order to allow a successful decoding of the WZ frames at feasible WZ bit rates. This condition can result in a very high bit rate requirement, which is not possible in limited-bandwidth applications. Additionally, when the key frames are too distant apart, the quality of the side information is degraded. As a result, most research on DVC considers a group of pictures (GOP) size of 2, that is, each key frame is followed by one WZ frame. Several attempts have been made to increase the GOP size in DVC. In [16], Aaron et al. impose the use of high-quality key frames with fixed GOP sizes ranging from 2 to 5. As the GOP size increases, the system's performance decreases. However, lower rates could be reached with greater GOP sizes due to the high bit rate requirements of the key frames. In [17], Ascenso et al., present a content-adaptive GOP size selection algorithm. The number of frames in a GOP is determined dynamically depending on motion activity. However, the proposed algorithm uses four different metrics in order to determine the size of a GOP, which results in a significant increase of the encoder's complexity. Furthermore, both studies rely on a feedback channel for the decoding of WZ frames, and on H.263+ for key frame encoding. Since H.264/AVC [23] greatly outperforms H.263+, it is expected that H.264 intracoding will outperform both Wyner-Ziv systems too.

In this paper, we present novel algorithms for dynamically varying the GOP size in distributed video coding. Our simulations are performed using a pixel-domain WZ video codec. However, the same algorithms can be applied in a transform-domain codec, which improves the overall performance at the expense of a slight increase in encoding and decoding complexity. Our method relies on H.264 for the encoding of key frames, and on our previously developed WZ rate estimation technique presented in [5, 6, 14, 15], where quadri-binary turbo-codes are used for the compression of WZ frames. Only one metric is required to determine the size of a GOP, and a feedback channel is not needed for the decoding of WZ frames. Automatic mode selection allows the system to switch to H.264 intracoding mode in regions where H.264 outperforms WZ video coding. Furthermore, based on our study in [5, 6, 14, 15], our algorithms can be easily extended to take into account channel impairments and multiuser scenarios.

This paper is organized as follows: in Section 2, we present a brief description of the Wyner-Ziv video codec used in this study, along with the rate estimation technique for WZ frames. Our adaptive algorithms for GOP size control are detailed in Section 3, and the additional complexity they incur at the encoder side is analyzed in Section 4. Finally, simulation results are presented in Section 5.

2. Description of the Distributed Video Coding System

The distributed video coding system considered in this study can be represented by the block diagram in Figure 1. Key frames are compressed using H.264 intracoding. After H.264 decoding, a key frame is stored in a buffer in order to be used during the process of generating the side information necessary for the decoding of WZ frames.

Compression of the Wyner-Ziv frames starts by a uniform scalar quantization to obtain $M$-bit representations of the eight-bit pixels, $M \in \{1, 2, 4\}$. The turbo encoder
[24–27] consists of a parallel concatenation of two 16-state quadri-binary convolutional encoders separated by an internal interleaver and resulting in a minimum global coding rate of 2/3. The generator polynomials in octal notation are \((23, 35, 31, 37, 27)_8\) from [27]. At the encoder output, systematic information is discarded, while parity information is punctured and transmitted to the decoder. Side information (SI) of a particular WZ frame is generated by motion-compensated interpolation of two previously reconstructed (WZ or H.264) frames. The frame interpolation technique assumes symmetric motion vectors as explained in [8, 16]. The interpolated frame is then quantized and fed to the turbo decoder as a noisy version of the missing systematic data. Turbo decoding is realized by iterative Soft Input Soft Output (SISO) decoders based on the Max-Log-MAP (maximum a posteriori) algorithm [28]. However, metric calculations are modified in order to take into account the nonbinary nature of the turbo codec, and the residual signal statistics between the WZ and SI frames.

Finally, the reconstruction block is used to recover an eight-bit version of the decoded WZ frames using the available side information [8]. The final output is then stored in a buffer if needed to generate side information for another WZ frame.

In a previous work [5], we presented an analytical approach for estimating the compression limits of a pixel-domain Wyner-Ziv video coding system with a transmission over error-prone channels, without the need for a feedback channel. Simulation results showed that the theoretical bounds can be used in a broadcasting system to predict the compression level for each frame with a minor loss in the decoding PSNR, compared to the classical feedback-based coding system. In the absence of transmission errors, the theoretical compression bound of the WZ frames for the system in Figure 1 can be expressed as (see [5, 6])

\[
H(X|Y) = -\frac{1}{2^M} \sum_{i=0}^{2^M-1} \sum_{j=0}^{2^M-1} g \left( \frac{\alpha}{2} \frac{2^M e^{-\alpha|d_{i-j}|}}{L_{d_{i-j}}} \right),
\]

where \(X\) is a quantized WZ frame, \(Y\) is the corresponding side information, \(g(a) = a \log_2(a)\), \(L_{d_{i-j}} = 2^M - |i - j|\) is the number of possible couples \((i, j)\) that yield the difference \(i - j = d\), \(d_{i-j} = 2^M(i - j)\) is the difference between two quantized pixel values, \(\alpha\) is a scaling factor, and \(\alpha\) is the parameter of the Laplacian distribution modeling the statistics of the residual error between the side information and the WZ frame [8]. Since there is always a gap between theoretical and practical bounds, we have determined in [14] that, in order to obtain a good average performance, the ratio between the average number of transmitted bits per pixel and the lower compression bound expressed in (1) must be not less than \(T_4 = 2.4\) for \(M = 4\), \(T_2 = 4\) for \(M = 2\), and \(T_1 = 6\) for \(M = 1\). As a result, the encoder can determine the compression rate for a given WZ frame by first determining its compression bound, then multiplying it with the corresponding coefficient \(T_M\), depending on the value of the WZ quantization parameter \(M\).
3. Adaptive Algorithms for GOP Size Control in Wyner-Ziv Video Coding

In a video sequence, when there is low motion, consecutive frames are highly correlated. The aim of varying the GOP size is to allow the system to better exploit this property, by reducing the number of intracoded key frames in regions where WZ frames would yield a better rate-distortion (R-D) performance. In regions where intracoding outperforms WZ coding (because of high motion), the GOP structure is reduced to one (H.264 intracoded) frame per GOP. This automatic mode selection allows the WZ encoder to make use of H.264 coding efficiency to better improve the system’s R-D performance.

Let \( S_{\text{max}} \) represent the maximum allowable GOP size. For each GOP, let \( R_0 \) represent the average bit rate assigned for the first frame (intracoded key frame) in the GOP, and \( \text{PSNR}_0 \) its PSNR. \( S_{\text{max}} \) can be chosen depending on the system’s delay constraints. For a GOP of size \( S \), let \( F_0 \) denote the key frame, \( F_1, F_2, \ldots, F_{S-1} \) the WZ frames, and \( F_S \) the key frame of the next GOP.

To perform GOP length decision, our proposed algorithm operates as follows:

Initially, set \( S = 1 \).

While \( S \leq S_{\text{max}} \) do:

If \( S = 1 \), go to step (e), otherwise:

(a) Interpolate between \( F_0 \) and \( F_S \)

The interpolated frame serves as an estimate of the side information available at the decoder during the decoding process of the WZ frame \( F_{(S/2)} \), located at half-distance between the two frames.

Since motion estimation is not allowed at the encoder for complexity reasons, average interpolation [8] can be used to estimate the side information that will be available at the decoder.

(b) Estimate the average bit rate \( R_{(S/2)} \)

Given the WZ frame \( F_{(S/2)} \) and its corresponding side information (estimate), the encoder determines its lower compression bound, and consequently, its compression rate, as explained in the previous section. The computation of \( R_{(S/2)} \) becomes straightforward.

(c) Compute \( \text{PSNR}_{(S/2)} \)

Given the WZ frame \( F_{(S/2)} \) and its corresponding side information (estimate), the encoder can determine an estimate \( F'_{(S/2)} \) of the decoded frame at the receiver by first quantizing the WZ frame, and then reconstructing an 8-bit version using the available side information. The PSNR is then computed using \( F_{(S/2)} \) and \( F'_{(S/2)} \).

(d) Repeat steps (a) to (c) until rate and PSNR estimates are obtained for all the frames of the GOP

However, instead of interpolating between \( F_0 \) and \( F_S \) in step (a), the frames \( F_0 \) and \( F'_{(S/2)} \) are first used to generate a side information estimate for the frame located at half distance between the two, and the same process in steps (b) to (d) is repeated. Then, a similar procedure is performed in the second half of the GOP, using the frames \( F'_{(S/2)} \) and \( F_S \). Frame \( F_{(S/2)} \) is used instead of \( F_{(S/2)} \) because the former better estimates the frame that will be available at the decoder side since the latter is not known by the decoder.

(e) Estimate the average rate and PSNR obtained with a GOP of size \( S \), respectively, defined as

\[
R^S_{av} = \frac{1}{S} \sum_{j=0}^{S-1} R_j, \quad \text{PSNR}^S_{av} = \frac{1}{S} \sum_{j=0}^{S-1} \text{PSNR}_j.
\]

(f) Determine \( \lambda^S_{av} = \text{PSNR}^S_{av}/R^S_{av} \)

This represents the average PSNR per average unit bit rate estimated for a GOP of size \( S \).

(g) \( S = S + 1 \).

Intuitively, the best R-D performance is obtained by maximizing the average PSNR per unit bit rate. As a result, the system decides the GOP length \( L \) as

\[
L = \arg \max_{k=1,2,\ldots,S_{\text{max}}} (\lambda^k_{av}).
\]

In other words, if the system determines that the average PSNR per unit bit rate obtained by WZ coding, for different GOP lengths \( S > 1 \), is lower than the one obtained with H.264 intracoding \( S = 1 \), the system switches to H.264 intracoding mode \( L = 1 \) and an H.264 I-frame is then transmitted. Otherwise \( L > 1 \), an H.264 intracoded key frame is transmitted, followed by \( L - 1 \) WZ frames. Furthermore, since motion-compensated interpolation yields better side information compared to average interpolation, in general, the decoder is expected to perform better than estimated at the encoder side. This procedure is repeated at the beginning of every GOP and thus, the GOP length is dynamically varied along the sequence, in order to optimize the overall performance.

Figure 2 presents the algorithm above as a flow chart diagram, and Pseudocode 1 shows the Pseudocode of the recursive procedure \( \text{R}_{\text{PSNR}}, \text{estimations}(i, \text{len}) \) used to estimate the rate and PSNR for all the frames in a GOP of size \( S \), where \( i \) is the time index of the first frame in the GOP, and \( \text{len} \) is the time interval between the frame at index \( i \) and the next frame used during the interpolation process (initially, \( \text{len} = S \)).

In general, a constant quality is desired along the sequence, since big fluctuations in PSNR yield undesirable visual effects. For this reason, the encoder determines the rate for the H.264 intracoded frames in such a way to obtain a near-constant PSNR in the GOP. This can be done by one of several techniques:

(i) Using predefined performance tables that determine H.264 rate-distortion relationships.
(ii) Using an analytical model for H.264 rate-distortion performance. This allows the system to avoid extensive table search as in the previous technique. However, it is important in this case to have an accurate, generalized R-D model [29].

(iii) Trial and error: the system tries several coding rates for each H.264 intracoded frame, and determines the PSNR for each case. Then, the rate that yields a PSNR closer to the one of neighboring frames is chosen. This method is more accurate than the previous ones. However, it can result in significant delay and increased encoder complexity.

The GOP size control algorithm can be further simplified by assuming a constant PSNR for all the frames. In this case, instead of maximizing the average PSNR per unit bit rate, (3) reduces to minimizing the average bit rate per frame over all possible GOP sizes. As a result, the simplified GOP size control algorithm operates as follows:

Initially, set $S = 1$.

While $S \leq S_{\text{max}}$ do:

- If $S = 1$, go to step (d), otherwise:
  - (a) Interpolate from $F_0$ and $F_5$.
  - (b) Estimate the average bit rate $R_{\text{av}}(S/2)$.
  - (c) Repeat steps (a) and (b) until a rate estimate is obtained for all the frames in the GOP (replace frames $F_0$ and $F_5$ in step (a) with the corresponding frames as previously explained in step (d) of the initial algorithm).
  - (d) Determine $R_{\text{av}}^S = (1/S)\sum_{j=0}^{S-1} R_j$
  - (e) $S = S + 1$

Finally, the system decides the GOP length $L$ as

$$L = \arg \min_{k=1,2,...,S_{\text{max}}} (R_{\text{av}}^k).$$

This allows the system to avoid estimating the PSNR for each frame, and the average PSNR per unit bit rate for each GOP size.

4. Complexity Analysis

While the aim of DVC is mainly permitting the design of low-complexity encoders, our GOP size selection algorithms incur additional encoding complexity. In this section, we present an analysis of the proposed algorithms computational load, and we compare our dynamic algorithms with the one presented in [17].

Table 1 presents an estimation of the necessary number of additional operations (OPs) incurred by each iteration (for each frame in the GOP, for every GOP size $k, 1 \leq k \leq S_{\text{max}}$) in the initial (nonsimplified) GOP size control algorithm. $P \times Q$ represent the frame dimensions, and $M$ the quantization parameter for WZ frames.

Consider for example the number of operations performed to compute the PSNR. The calculation of the PSNR consists of first computing the mean square error (MSE), taking its inverse, multiplying it with a constant, computing the log of the result, and finally multiplying it by 10. The square error between two pixel values requires one addition operation and one multiplication. To compute the MSE, $PQ$ square errors are first computed (which results in $PQ$ additions and $PQ$ multiplications) and summed together ($PQ - 1$ additions). The final result is then divided by $PQ$. As a result, $2PQ - 1$ additions, $PQ + 4$ multiplications, and a log operation are performed in order to obtain the PSNR. A similar analysis was performed on the other operations involved in the GOP size control algorithm, and the results are summarized in Table 1.

The total number of operations is roughly obtained by summing the elements of the last row in the table, which results in $12PQ + (8 \times 2^M) + 3$ operations. Since, in our codec, the maximum value for $M$ is 4, and by assuming QCIF video sequences ($P \times Q = 144 \times 176$), the total number of operations becomes 306,179 OPs. In the simplified algorithm, the computation of the PSNR is not performed. Thus, a reduction of $3PQ + 3$ operations is obtained, which yields a total number of 230,144 OPs.

A similar study was performed on the algorithm presented in [17], where four different metrics were used: the difference of histograms (DH), the histogram of difference (HD), the block histogram difference (BHD) and the block variance difference (BVD). Given the parameters specified in [17], we obtain 50,784 OPs for DH, 50,736 OPs for HD, 60,191 OPs for BHD, and 161,567 OPs for BVD, which results in a total of 323,278 OPs.

Even though the computational load of our initial algorithm and the one of the algorithm in [17] are almost similar, it is important to note that our algorithm presents the additional property of estimating the necessary bit rate without the need for a feedback channel, and the possibility to take into account channel impairments based on our study in [5, 6, 14, 15]. On the other hand, a reduced complexity of approximately 25% can be obtained with our simplified algorithm, without a significant loss in R-D performance as will be shown in the next section.

5. Experimental Results

In our simulations, we consider three different QCIF video sequences with different levels of motion: Foreman, Grandmother, and Salesman, sampled at a rate of 30 frames per second. The first 100 frames from each sequence are first encoded using a WZ codec with fixed GOP sizes ranging from 1 to 5. For the case where the GOP size is 1, all frames are H.264 intracoded, whereas for the other cases, only the first frame (key frame) from each GOP is H.264 intracoded, whereas for the other frames, only the first frame (key frame) from each GOP is H.264 intracoded, while the remaining ones are WZ-coded. H.264 coding is performed using JM FRExt reference software, version 13.2, with baseline profile. The results are then compared with the case where a WZ codec with a dynamically varying GOP size.
procedure R\_PSNR\_estimations(i, len) {
    If len < 2
        Return // End of the recursive function calls
    Else
        a = i, b = i + len // a and b are the time indices
        // of the frames used during the
        // interpolation process.
        d = ⌊(b - a)/2⌋ // time interval from a to the
        // frame at mid distance
        // between a and b.
        SI_{[a+d]} = Interpolate(a, b) // Perform average interpolation
        // between frames at time
        // indices a and b.
        Q_{[a+d]} = QuantizeFrame(F_{[a+d]})
        R_{[a+d]} = EstimateBitrate(Q_{[a+d]}) // estimate the bitrate as
        // explained in Section 2.
        F’_{[a+d]} = Reconstruct(Q_{[a+d]}, SI_{[a+d]}) // reconstruct the
        // quantized WZ frame
        // given the estimated
        // side information.
        PSNR_{[a+d]} = ComputePSNR(F’_{[a+d]}, F_{[a+d]}) // Compute the PSNR
        // of the reconstructed
        // frame.
        R\_PSNR\_estimations(i, d); // Recursive function call using
        // the first half of the GOP.
        R\_PSNR\_estimations(i + d, len − d) // Recursive function call using
        // the second half of the GOP.
    End If
}

PSEUDOCODE 1: Pseudocode of the recursive procedure R\_PSNR\_estimations used to estimate the rate and PSNR for all the frames in a GOP.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Average Interpolation</th>
<th>Reconstruction</th>
<th>Quantization (WZ and SI)</th>
<th>Estimation of the α parameter</th>
<th>PSNR estimation</th>
<th>Rate (R) estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additions</td>
<td>PQ</td>
<td>PQ</td>
<td>2PQ − 1</td>
<td>2PQ − 1</td>
<td>1 × 2^{2M} − 1</td>
<td></td>
</tr>
<tr>
<td>Multiplications</td>
<td>PQ</td>
<td></td>
<td>PQ + 1</td>
<td>P + 4</td>
<td>5 × 2^{2M} + 1</td>
<td></td>
</tr>
<tr>
<td>Log or Exp</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2 × 2^{2M}</td>
<td></td>
</tr>
<tr>
<td>Comparisons</td>
<td>PQ</td>
<td></td>
<td></td>
<td>2PQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical AND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>2PQ</td>
<td>2PQ</td>
<td>3PQ</td>
<td>3PQ + 3</td>
<td>8 × 2^{2M}</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Number of additional operations per frame for each GOP size k, 1 ≤ k ≤ S_{max}, incurred by the proposed GOP size control algorithm.

size is used. The GOP size is determined using our proposed algorithms as explained in Section 3, with S_{max} set to 5.

In Figure 3, we show the rate and PSNR variations along the Grandmother sequence for M = 4, and the quantization parameter of the H.264 intraframes QP = 25, obtained using a WZ codec with a fixed GOP size set to 3, with and without a feedback channel (FC). It can be noticed that the rate estimated without FC exceeds the rate obtained using FC most of the time. As a result, WZ frames are correctly decoded in both cases and the reconstructed output is the same. However, in rare situations (e.g., in frame 41), the encoder underestimates the rate needed for correctly decoding a WZ frame, which yields a degraded quality at the decoder output. For an average bit rate of 697 kbps
obtained with the feedback channel, an average rate excess of 50 kbps is observed in the case where the feedback channel is suppressed. In other words, for applications where the feedback channel is not suitable (e.g., in video broadcasting), the cost to be paid (in terms of bit rate) for the suppression of the return channel is approximately 7%.

In Figure 4, we show the average $R$-$D$ curves obtained for the three sequences using both the initial (I) and simplified (S) algorithms. The rate and PSNR are averaged over all the sequence (key and WZ frames). Different rate points are obtained by varying the quantization parameter $M$ for the WZ frames. As for the quantization parameter (QP) of the H.264 intraframes, it is chosen in such a way to permit a near-constant decoding quality in the output video sequence, using the approach presented in [17]. It can be clearly seen that, for the Salesman and Foreman sequences, both curves overlap (both algorithms have similar performance), whereas a negligible loss that does not exceed 0.45 dB is observed with the Grandmother by using the simplified algorithm.

Figures 5 to 7 show the average $R$-$D$ performance for the Foreman, Grandmother, and Salesman sequences, respectively, obtained with the initial (nonsimplified) algorithm. In Figure 5 (Foreman), we notice that for the case of a fixed GOP size, the performance decreases as the GOP size increases. The best performance is thus obtained when all frames are intracoded. This is due to the high motion in this sequence, which yields less accurate side information when the key frames are further apart. A similar effect has been noticed in [16] where key frames were encoded using an H.263+ video codec. However, when the GOP size is dynamically varied along the sequence, a gain of 10 to 12 kbps is obtained compared to H.264 intracoding. For sequences with lower motion levels, different results are observed. It can be seen in Figures 6 and 7 that the best system performance is obtained with a GOP of size 3 (for the fixed-GOP case). Our proposed system outperforms both H.264 intracoding and fixed-GOP WZ coding in most cases. For example, for the Grandmother sequence at 520 kbps, a gain of 3 dB is observed with respect to the H.264 intracodec and 0.8 dB with respect to the WZ codec with a GOP size of 3. Similarly, for the Salesman sequence at 580 kbps, our proposed algorithm outperforms the H.264 intra codec and the WZ codec with a GOP size of 3 by 1 dB and 0.1 dB, respectively. However, a performance loss of 0.4 dB
can be observed with the Salesman sequence at 900 kbps using a dynamic GOP size, compared to the fixed-GOP (size = 3) WZ codec. In fact, this is due to a significant mismatch between the side information available at the encoder (estimated using average interpolation) and the one available at the decoder (obtained by motion-compensated interpolation).

Figures 8 and 9 show the rate and PSNR variations along the Salesman sequence for $M = 4$ and $M = 2$, respectively, for the case where the rate estimation is done at the encoder using average-interpolated side information, and for the case where this estimation is performed at the decoder using the side information obtained by motion-compensated interpolation. The source coding rate is the one estimated by the encoder, whereas the real PSNR is the one obtained after the decoding process and thus, the corresponding curves are shown as solid lines. On the other hand, decoder-estimated bit rate and encoder-estimated PSNR (dotted curves) are shown only to analyze the system’s behavior at both (encoder and decoder) sides. It can be seen that, in some regions (e.g., frames 61 to 64 and frames 71 to 74 with $M = 4$), the encoder underestimates the rate necessary for the decoding of WZ frames, which yields a very high bit error rate at the turbo decoder output. As a result, the reconstruction function of the WZ codec cannot yield a reliable output and thus, a significant performance loss is observed in these regions, which greatly affects the average system performance as shown in Figure 7. However, such estimation errors rarely occur. As it can be clearly seen in Figures 8 and 9, the encoder estimations are accurate most of the time when $M = 4$, and all the time when $M = 2$. Similar results were observed with the other sequences for different values of $M$.

Table 2 shows the percentage of GOP sizes obtained for each of the sequences using the proposed adaptive GOP size control algorithm (nonsimplified), for different values of the WZ quantization parameter $M$. For the Foreman sequence, when $M = 1$, 100% of the GOPs are of size 1. In other words, the system switches to H.264 intracoding mode all the time, and no frame in the sequence is WZ-encoded. For $M = 2$ and 4, most of the GOPs are of size 1, while the maximum GOP size does not exceed 3. This explains the reason behind the similar performance between the H.264
Figure 9: Rate (a) and PSNR (b) variations along the Salesman sequence using a WZ codec with the proposed GOP size control algorithm for $M = 2$.

Table 2: Percentage of GOP sizes used in each sequence.

<table>
<thead>
<tr>
<th>Gop size</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>$M = 1$</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$M = 2$</td>
<td>91.1</td>
<td>6.7</td>
<td>2.2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$M = 4$</td>
<td>87.4</td>
<td>10.3</td>
<td>2.3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$M = 1$</td>
<td>0</td>
<td>4</td>
<td>20</td>
<td>36</td>
</tr>
<tr>
<td>Grandmother</td>
<td>$M = 2$</td>
<td>% 0</td>
<td>4.2</td>
<td>16.7</td>
<td>37.5</td>
</tr>
<tr>
<td></td>
<td>$M = 4$</td>
<td>23.1</td>
<td>33.3</td>
<td>20.5</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td>$M = 1$</td>
<td>3.7</td>
<td>22.2</td>
<td>18.5</td>
<td>0</td>
</tr>
<tr>
<td>Salesman</td>
<td>$M = 2$</td>
<td>18.2</td>
<td>12.1</td>
<td>39.4</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$M = 4$</td>
<td>22.2</td>
<td>7.4</td>
<td>3.7</td>
<td>0</td>
</tr>
</tbody>
</table>

and our proposed WZ codec for the Foreman sequence, as shown in Figure 5. More GOP size variations can be noticed with the two other sequences. For the Grandmother sequence with $M = 1$ and $M = 2$, the system is always in WZ coding mode. In other words, not any GOP is of size 1 and only key frames are intracoded, since the WZ coding outperforms H.264 intracoding in this case, according to the encoder estimations.

The Foreman sequence is characterized by very high motion levels, especially in its second half. In such regions with high motion, H.264 intracoding usually outperforms WZ coding. For this reason, in order to analyze our system’s performance with high motion video, we encoded the complete Foreman sequence (400 frames) and the results are reported in Figure 10. This figure shows the GOP size variations along the sequence for $M = 2$ and $M = 4$. Long runs of consecutive GOP sizes equal to 1 can be noticed in high motion areas, which indicate that the system has switched to H.264 intracoding mode in these regions. As a result, the R-D performance for the Foreman sequence is slightly better than the one obtained with a pure H.264 intracoder, as was also noticed in Figure 5.

6. Conclusion and Future Work

This paper presents simple algorithms that dynamically adapt the GOP size for a distributed Wyner-Ziv video codec, depending on the content of the video scene to be encoded. Based on H.264 intracoding for key frames, the system relies on theoretical calculations to estimate the bitrate necessary to successfully decode Wyner-Ziv frames without the need for a feedback channel, which makes it suitable for broadcasting applications. Automatic mode selection allows the system to switch between H.264 intracoding and WZ coding modes in order to optimize the overall system performance. Simulation results show an average gain that can reach 3 dB compared to an H.264 intracodec and 0.8 dB compared to a WZ codec with a fixed GOP size.

As a future work, authors will focus on taking into account the transmission channel conditions in the GOP size control algorithm, and implementing the system in a more realistic network environment with multiple users, based on their previous research in [14, 15]. Further research may consider dynamically varying the quantization parameters (QP for key frames and $M$ for WZ frames) and using advanced interpolation techniques to improve the overall performance.
Acknowledgment

This work has been supported by a research grant from the Lebanese National Council for Scientific Research (LNCSR).

References


Submit your manuscripts at http://www.hindawi.com